

Guide - PRISSMA

13 avril 2010

Table des matières

1	Le code	5
1.1	Introduction	5
1.2	Historique	5
1.3	Les bases théoriques	7
1.3.1	Angular discretization	8
1.3.2	Spatial discretization for hybrid grids	8
1.3.3	Spectral gas properties	10
2	Installer le code	17
2.1	Configuration système requise	17
2.1.1	Logiciels requis	17
2.1.2	Logiciels recommandés	18
2.1.3	Aide à l'installation	18
2.2	Téléchargement du code source	19
2.3	Organisation du code	20
2.4	Installation	20
2.4.1	Variables d'environnement	20
2.4.2	Définition de son host	21
2.5	Compilation	22
2.5.1	Compilation des outils	22
2.5.2	Compilation du code séquentiel	22
2.5.3	Compilation du code parallèle	23
3	Préparation et lancement d'un calcul	25
3.1	Présentation d'un exemple de cas test	25
3.2	Préparation du calcul	25
3.2.1	Création du répertoire de travail	25
3.2.2	Lecture du maillage	27
3.2.3	Lecture de la solution initiale	28
3.2.4	Lecture des conditions aux limites	30
3.2.5	Conversions des fichiers d'entrée	31

3.3	Lancement du calcul	33
3.4	Traitement des résultats	33
3.4.1	Création de fichiers ascii sur un rayon	33
3.4.2	Création de fichiers compatibles avec ParaView	34
4	Utilisation des tests	37
4.1	Téléchargement	37
4.2	Présentation des configurations utilisées	37
4.2.1	le cube	37
4.2.2	le cylindre homogène et isotherme	38
4.2.3	le cylindre inhomogène et anisotherme	38
4.3	Les tests pour l'utilisateur	39
4.3.1	Tester son installation	39
4.3.2	Influence du modèle de discrétisation spatiale	40
4.3.3	Influence de la quadrature	41
4.3.4	Influence du modèle spectral	42
4.3.5	Remise à zéro	43
4.4	Les tests pour le développeur	45
4.4.1	Validation numérique	45

Chapitre 1

Le code

1.1 Introduction

PRISSMA (Parallel RadIation Solver with Spectral integration on Multicomponent mediA) est un code de calcul de transfert radiatif développé en Fortran. Il permet de résoudre l'équation de transfert radiatif pour calculer la luminance dans une géométrie complexe à partir de solutions issues de calcul de CFD ¹ : les champs de concentration et de température.

Le calcul de la luminance I permet d'obtenir :

- G , le rayonnement incident ;
- H , l'éclairement surfacique ;
- S_r , le terme source radiatif ;
- Q , le flux aux parois.

1.2 Historique

Le code PRISSMA s'appuie sur une longue collaboration entre Mouna El Hafi du Centre Énergétique et Environnement de l'École des Mines d'Albi-Carmaux ² et Bénédicte Cuénot au CERFACS (Toulouse) ³, voir Fig. 1.1. Dans sa version actuelle, le code PRISSMA a été écrit par Jorge Amaya (thèse au CERFACS 2006-2010) et développé conjointement avec Damien Poitou (thèse École des Mines d'Albi/CERFACS 2006-2009).

¹Computational Fluid Dynamics

²<http://www.mines-albi.fr>

³Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique : <http://www.cerfacs.fr>

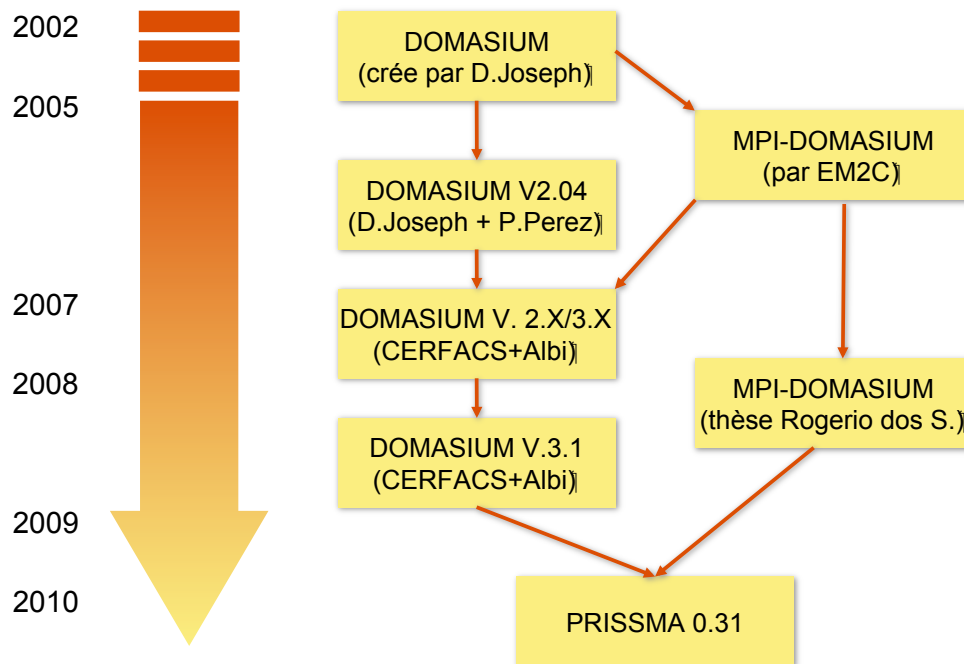


FIG. 1.1 – Historique du code PRISSMA.

Initialement un premier code a été développé durant la thèse de David Joseph[1] (2001-2004) : DOMASIUM (Discrete Ordinates Method Applied with Spectral Integration on Unstructured Meshes). Ce code a permis de mettre en place un code radiatif adapté aux maillages non structurés utilisés en CFD dans le but de réaliser des calculs de transferts radiatifs dans des configurations de combustion. Ce code est basé sur la méthode des ordonnées discrètes dans un souci d'efficacité de temps de calcul, cette méthode offrant un bon compromis entre précision et coût de calcul. Cette méthode est utilisée conjointement avec un modèle spectral en k -distributions sur des bandes étroites du spectre infrarouge. Ce modèle permet une description précise des propriétés d'absorption du milieu composé de gaz produits de combustion (H_2O , CO_2 , CO) et éventuellement de suies. La première version du code est principalement séquentielle bien qu'une première tentative de parallélisation ait été réalisée par D. Joseph.

À partir de la version originale de DOMASIUM, différents développements ont été réalisés dans le but d'optimiser le code dans le cadre du projet CORAYL (PROJET ANR-05-CICG-012). Ce projet est une collaboration entre l'École des Mines d'Albi, le CERFACS et le laboratoire EM2C⁴ de

⁴Laboratoire d'Énergétique Moléculaire et Macroscopique, Combustion

l'École Centrale Paris. David Joseph, Patrice Perez et Damien Poitou ont réalisé dans ce projet un étude sur la mise en place de modèles spectraux rapides pour réduire les temps de calculs radiatifs. Durant la thèse de Rogerio Dos Santos une version optimisée et parallélisée a été développée à l'EM2C : MPI-DOMASIUM. Cette version a permis de réduire le temps de calcul du code et de démontrer la faisabilité d'un couplage rayonnement-combustion turbulente avec le code AVBP⁵.

Dans le but de synthétiser les différentes versions précédentes pour les intégrer dans un code intrinsèquement parallèle, un nouveau code a entièrement été réécrit : PRISSMA. Ce code a été optimisé en mémoire afin de rechercher les meilleures performance de calcul dans un contexte de calculs parallèles. Ce travail de réécriture du code a été principalement réalisé par J. Amaya. De nouveaux modèles spectraux ont également été intégrés durant la thèse de D. Poitou[2] ce qui a permis d'atteindre des rapports de temps de calcul PRISSMA/AVBP inférieurs à 1 montrant la faisabilité de couplage dans des configurations industrielles. Durant le développement de PRISSMA la stabilité et l'utilisation du code ont été grandement améliorés notamment par la création d'un ensemble d'outils de pré et post-processing.

Nous allons rappeler dans la partie suivante les bases théoriques sur lesquelles s'appuient le code.

1.3 Les bases théoriques

PRISSMA [3, 4] has been designed to simulate the radiative heat transfer in coupled simulations with flow dynamics, involving unstructured grids. In the following and for sake of clarity, the intensities and radiative properties are expressed for a single wavenumber (monochromatic case) but the formulation can be easily extended to a full spectrum.

Discrete Ordinates Method have been introduced first by [5] and have been widely used in radiative transfer applications. Considering an absorbing-emitting and non-scattering gray medium, the variation of the radiative intensity $I(\mathbf{s})$ along a line of sight can be written as :

$$\frac{dI(\mathbf{s})}{ds} = \kappa I_b - \kappa I(\mathbf{s}) \quad (1.1)$$

where $I(\mathbf{s})$ is the radiative intensity along the directional coordinate \mathbf{s} , I_b is the blackbody radiative intensity, and κ is the absorption coefficient. Boundary conditions for diffuse surfaces are taken from the relation giving the

⁵<http://www.cerfacs.fr/4-26334-The-AVBP-code.php>

intensity leaving the wall I_w as a function of the blackbody intensity of the wall $I_{b,w}$ and of the incident radiative intensity :

$$I_w(\mathbf{s}) = \epsilon_w I_{b,w} + \frac{\rho_w}{\pi} \int_{\mathbf{n} \cdot \mathbf{s}' < 0} I_w(\mathbf{s}') |\mathbf{n} \cdot \mathbf{s}'| d\Omega' \quad (1.2)$$

where ϵ_w is the wall emissivity, ρ_w the wall reflectivity, \mathbf{n} the unit vector normal to the wall and \mathbf{s}' the direction of propagation of the incident radiation confined within a solid angle $d\Omega'$.

1.3.1 Angular discretization

In the DOM, the calculation of a radiative source term at a given point is based on the discretization of the Radiative Transfer Equation (Eq.1.1) according to a chosen number N_{dir} of discrete directions $\mathbf{s}_i(\mu_i, \eta_i, \xi_i)$, associated to the corresponding weights w_i , contained in the solid angle 4π , and where (μ_i, η_i, ξ_i) are directional cosines. Different angular discretizations may be used. In a recent study, Koch and Becker [6] compare the efficiency of several types of angular quadratures. They recommend the LC_{11} quadrature for its better accuracy. However calculations performed with the S_4 quadrature satisfy a good compromise between accuracy and rapidity as shown in [4], and may also be used.

1.3.2 Spatial discretization for hybrid grids

The RTE (Eq.1.1) is solved for every discrete direction \mathbf{s}_i using a finite volume approach. The integration of the RTE over the volume V of an element limited by a surface Σ , and the application of the divergence theorem yields :

$$\int_{\Sigma} I(\mathbf{s}_i) \cdot \mathbf{s}_i \cdot \mathbf{n} d\Sigma = \int_V (\kappa I_b - \kappa I(\mathbf{s}_i)) dV \quad (1.3)$$

The domain is discretized in three-dimensional control volumes V . It is assumed that I_b and $I(\mathbf{s}_i)$ are constant over the volume V and that the intensities I_j at the faces are constant over each face. Considering that I_j is the averaged intensity over the j^{th} face, associated with the center of the corresponding face, that $I_{b,P}$ and I_P are the averaged intensities over the volume V , associated with the center of the cell, and assuming plane faces and vertices linked by straight lines, Eq.(1.3) can be discretized as follows :

$$\sum_{j=1}^{N_{face}} I_j(\mathbf{s}_i) \cdot (\mathbf{s}_i \cdot \mathbf{n}_j) A_j = \kappa V (I_{b,P} - I_P(\mathbf{s}_i)) \quad (1.4)$$

where \mathbf{n}_j is the outer unit normal vector of the surface j .

The scalar product of the i^{th} discrete direction vector with the normal vector of the j^{th} face of the considered cell is defined by D_{ij} :

$$D_{ij} = \mathbf{s}_i \cdot \mathbf{n}_j = \mu_i n_{xj} + \eta_i n_{yj} + \xi_i n_{zj} \quad (1.5)$$

The discretization of the boundary condition (Eq.(1.2)) is straightforward :

$$I_w = \epsilon_w I_{b,w} + \frac{1 - \epsilon_w}{\pi} \sum_{\mathbf{n} \cdot \mathbf{s}_i < 0} w_i I(\mathbf{s}_i) |\mathbf{n} \cdot \mathbf{s}_i| \quad (1.6)$$

For each cell, the incident radiation G is evaluated as follows :

$$G = \int_{4\pi} I(\mathbf{s}) d\Omega \simeq \sum_{i=1}^{N_{dir}} w_i I(\mathbf{s}_i) \quad (1.7)$$

and the incident heat flux H_w at the wall surfaces is :

$$H_w = \int_{\mathbf{n} \cdot \mathbf{s} < 0} I(\mathbf{s}) |\mathbf{n} \cdot \mathbf{s}| d\Omega \simeq \sum_{\mathbf{n} \cdot \mathbf{s}_i < 0} w_i I_i |\mathbf{n} \cdot \mathbf{s}_i| \quad (1.8)$$

For a gray medium, the radiative source term S_r is given by :

$$S_r = \nabla \cdot Q_r = \kappa(4\pi I_b - G) \quad (1.9)$$

where Q_r is the radiative heat flux, and the radiative net heat flux at the wall is :

$$Q_w = \epsilon\pi I_{b,w} - H \quad (1.10)$$

For the evaluation of the radiative intensity $I(\mathbf{s}_i)$ in Eq. (1.6) to (1.10) Ströhle et al. [7] proposed a simple spatial differencing scheme based on the mean flux scheme that proved to be very efficient in the case of hybrid grids. This scheme relies on the following formulation :

$$I_P = \alpha \overline{I_{out}} + (1 - \alpha) \overline{I_{in}} \quad (1.11)$$

where $\overline{I_{in}}$ and $\overline{I_{out}}$ are respectively the intensities averaged over the entering and the exit faces of the considered cell. α is a weighting number between 0 and 1. Substituting $\overline{I_{out}}$ from Eq.(1.11) into Eq.(1.4) yields (for more details see [3]) :

$$I_P = \frac{\alpha V \kappa I_b - \sum_{\substack{j \\ D_{ij} < 0}} D_{ij} A_j I_j}{\alpha \kappa V + \sum_{\substack{j \\ D_{ij} > 0}} D_{ij} A_j} \quad (1.12)$$

The case $\alpha = 1$ corresponds to the Step scheme used by Liu et al. [8]. The case $\alpha = 0.5$ is called Diamond Mean Flux Scheme (**DMFS**) which is formally more accurate than the Step scheme. After calculation of I_P from Eq.(1.12), the radiation intensities at cell faces such that $D_{ij} > 0$ are set equal to $\overline{I_{out}}$, obtained from Eq.(1.11). For a given discrete direction, each face of each cell is placed either upstream or downstream of the considered cell center (a face parallel to the considered discrete direction plays no role). The control volumes are treated following a sweeping order such as the radiation intensities at upstream cell faces are known. This order depends on the discrete direction under consideration. An algorithm for the optimization of the sweeping order has been implemented [3]. Note that this sweeping order is stored for each discrete direction, and only depends on the chosen grid and the angular quadrature, i.e. it is independent on the physical parameters or the flow and may be calculated only once, prior to the full computation.

1.3.3 Spectral gas properties

The absorption coefficient κ of the combustion products is highly dependent on the wavenumber ν as shown by line spectra of radiative gases (H_2O , CO_2 and CO). To take into account this spectral dependency, the absorption coefficient of each species is here represented by the *SNB-ck* model [9, 10, 11]. For the gas mixture composed of different species, the same model is used, building data according to the mixing model exposed by Liu [11]. The radiative solutions are obtained by computing $N_{bands} \times N_{quad}$ independent calculations where $N_{bands} = 367$ is the number of narrow bands of spectral width $\Delta\eta = 25 \text{ cm}^{-1}$, describing the spectral properties in the range $150 - 9300 \text{ cm}^{-1}$, and $N_{quad} = 5$ is the number of the Gauss-Legendre quadrature points used for the spectral integration over each narrow band. For non-gray media, introducing spectral dependencies in Eq. (1.9), gives for the source term :

$$S_{r,DOM} = \sum_{i=1}^{N_{band}} \sum_{j=1}^{N_{quad}} \Delta\nu_i w_{ij} \kappa_{ij} (4\pi \bar{I}_{b,ij} - G_{ij}) \quad (1.13)$$

where G_{ij} is obtained from Eq.(1.7).

The computational efficiency of such a model is strongly linked to the number of bands N_{bands} , that has to be optimized depending on the studied case.

Other models called global models have been developed : WSGG (Weighted Sum of Gray Gas) [12] and FS-SNBcK (Full Spectrum SNBcK) [13].

Modèle SNB-CK

Le modèle SNB-CK est tout d'abord un modèle SNB et il considère donc un découpage du spectre des gaz rayonnants par des bandes dans lesquelles on fait l'hypothèse que la luminance du corps noir peut être considérée comme constante (hypothèse de bande étroite). Dans chaque bande étroite, les propriétés spectrales d'un gaz sont données représentées par deux paramètres $(\bar{\kappa}, \phi)$ tabulés pour des températures allant de 300 K à 2900 K par pas de 200 K, et pour des longueurs d'onde allant de 150 cm^{-1} à 9300 cm^{-1} . Les gaz considérés sont H_2O , CO_2 et CO et les données spectrales du modèles SNB proviennent de [9]. Ces données sont valables pour un modèle SNB basé sur un modèle de Malkmus pour lequel les intensités de raies d'un gaz dans chaque bande étroite sont distribuées suivant une loi inverse exponentielle [14, 15]. Avec de tels modèles, le transmittivité moyenne d'une colonne de gaz de longueur l homogène et isotherme pour une bande étroite de largeur $\Delta\nu$ est donnée par :

$$\bar{\tau}_{\Delta\nu}(l) = \exp \left[-\frac{\phi}{\pi} \left(\left(1 + \frac{2\pi\bar{\kappa}l}{\phi} \right)^{1/2} - 1 \right) \right] \quad (1.14)$$

Le calcul des grandeurs telles que le terme source radiatif ou le flux aux parois suppose une intégration sur le spectre en général et donc sur chaque bande étroite. Si on note κ_ν le coefficient d'absorption au nombre d'onde ν dans une bande étroite, le calcul d'une grandeur H intégrée sur la bande revient à calculer l'intégrale suivante :

$$\bar{H}_{\Delta\nu} = \frac{1}{\Delta\nu} \int_{\Delta\nu} H(\kappa_\nu) d\nu \quad (1.15)$$

Par conséquent, seule la valeur intégrée sur une bande étant importante, on a toute liberté de réorganiser les raies présentes dans une bande sans altérer le résultat et les calculs dans une bande. Autrement dit, seule la fonction de distribution des coefficients d'absorption dans la bande $f(\kappa)$ est importante, et pas la fréquence à laquelle ces coefficients d'absorption sont calculés.

La fonction de distribution des coefficients d'absorptions pour le modèle de Malkmus a une expression analytique [15] et est obtenue par :

$$f_{\Delta\nu}(\kappa) = \sqrt{\frac{\phi\bar{\kappa}}{2\pi\kappa^3}} \exp \left[-\frac{\phi}{2} \frac{(\kappa - \bar{\kappa})^2}{\kappa\bar{\kappa}} \right] \quad (1.16)$$

On utilise alors l'expression de la fonction cumulée de la distribution des coefficients d'absorption comme variable de description spectrale ($g(\kappa) =$

$\int_0^\kappa f(\kappa')d\kappa'$, qui a comme expression analytique pour un modèle de Malkmus :

$$g_{\Delta\nu}(\kappa) = \Gamma \left[-\sqrt{\frac{\phi}{\kappa/\bar{\kappa}}} \left(1 - \frac{\kappa}{\bar{\kappa}} \right) \right] + \exp(2\phi) \Gamma \left[-\sqrt{\frac{\phi}{\kappa/\bar{\kappa}}} \left(1 + \frac{\kappa}{\bar{\kappa}} \right) \right] \quad (1.17)$$

Par cette opération, l'intégrale précédente (Eq. 1.15) se réécrit :

$$\bar{H}_{\Delta\nu} = \frac{1}{\Delta\nu} \int_{\Delta\nu} H(\kappa_\nu) d\nu = \frac{1}{\Delta\nu} \int_0^1 H(\kappa(g)) dg \quad (1.18)$$

où g est la cumulative de la fonction de distribution du coefficient d'absorption. L'intégration d'une grandeur sur une bande étroite est effectivement réalisée en utilisant une quadrature sur la fonction g . La quadrature utilisée par défaut dans le code PRISSMA est une quadrature de Gauss-Legendre à 5 points [10], de sorte que l'intégrale précédente devient :

$$\bar{H}_{\Delta\nu} = \sum_{j=1}^{N_{quad}} w_j H(\kappa(g_j)) \quad (1.19)$$

où N_{quad} est le nombre de points de quadratures sur une bande étroite pour représenter la fonction g , et w_j est le poids associé au $j^{\text{ième}}$ point de la quadrature.

A ce stade, pour traiter des milieux non homogènes et non isothermes, on fait une hypothèse supplémentaire afin de prendre en compte la déformation du spectre le long d'un chemin optique. Cette hypothèse consiste à considérer que toutes les raies d'une bande étroite se déforment de la même manière en fonction de la température. Autrement dit, on peut considérer que la quantité $\frac{\kappa}{\bar{\kappa}}$ est constante quel que soit la fréquence dans la bande étroite. Cela revient également à considérer que la fonction g est constante le long d'un chemin optique. C'est l'hypothèse des K corrélés ou hypothèse CK.

Par ailleurs, le traitement du mélange de gaz est obtenu par le même modèle en suivant la méthode de Liu [11] basée sur la limite optiquement mince et qui donne les paramètres d'un mélange de gaz en fonction des paramètres de ses constituants :

$$\bar{\kappa}_{mélange} = \sum_{i=1}^{N_{gaz}} \bar{\kappa}_i \quad (1.20)$$

et

$$\frac{\bar{\kappa}_{mélange}^2}{\phi_{mélange}} = \sum_{i=1}^{N_{gaz}} \frac{\bar{\kappa}_i^2}{\phi_i} \quad (1.21)$$

Lors d'un calcul de terme source, les calculs radiatifs sont effectués $N_{bandes} \times N_{quad}$ fois de manière indépendantes, où $N_{bandes} = 367$ est le nombre de bandes étroites, de largeur $\Delta\eta = 25 \text{ cm}^{-1}$ considérées. Ces bandes permettent de considérer une région du spectre infrarouge allant de 150 cm^{-1} à 9300 cm^{-1} . $N_{quad} = 5$ représente le nombre de points de quadratures pour chaque bande étroite. L'équation du terme source radiatif calculé est donnée par :

$$S_{r,MOD} = - \sum_{i=1}^{N_{bandes}} \sum_{j=1}^{N_{quad}} \Delta\nu_i w_{ij} \kappa_{ij} \left(4\pi \bar{I}_{b,ij} - \sum_{k=1}^{N_{dir}} w_k I_{ij}(\mathbf{s}_k) \right) \quad (1.22)$$

Le modèle SNB-CK étant le modèle de propriétés radiatives des gaz disponible le plus précis dans PRISSMA, les résultats produits avec ce modèle serviront de résultats de référence pour les modèles FSCK et WSGG. Le code PRISSMA utilisé avec le modèle SNB-CK a fait l'objet de benchmarks avec d'autres codes et l'on pourra trouver des points de comparaison dans [4].

Modèle FS-SNBCK

En partant du modèle SNB-CK, Liu et al. [16] proposent d'opérer des regroupements de bandes étroites voisines et d'effectuer ensuite les calculs sur ces bandes élargies. Alors, pour un regroupement de M bandes étroites de même largeur, la transmittivité moyenne est donnée par :

$$\bar{\tau}_{wb} = \frac{1}{M} \sum_{i=1}^M \bar{\tau}_i \quad (1.23)$$

où $\bar{\tau}_i$ est la transmittivité moyenne de la i ème bande étroite dans le regroupement des M bandes. La transmittivité moyenne sur la bande large est alors simplement la moyenne des transmittivités moyennes des bandes étroites qui la composent. Si on considère maintenant une formulation en coefficients d'absorption, la fonction de distribution de la bande large est donnée par :

$$f_{wb}(\kappa) = \frac{1}{M} \sum_{i=1}^M f_i(\kappa) \quad (1.24)$$

où $f_i(\kappa)$ est la fonction de distribution des coefficients d'absorption dans la i ème bande étroite. La fonction cumulée peut être obtenue de la même façon par :

$$g_{wb}(\kappa) = \frac{1}{M} \sum_{i=1}^M g_i(\kappa) \quad (1.25)$$

Ce principe du regroupement de bandes introduit des erreurs dans l'hypothèse de bande étroite. En effet, plus on ajoute de bandes étroites voisines, et moins on peut considérer que la luminance du corps noir dans la bande est constante. Liu et al. [16] soulignent cette source d'erreurs et recommandent de se limiter dans le nombre de bandes regroupées. Cependant, il est possible de tenir compte de la variation de la luminance du corps noir dans les différentes bandes regroupées par une pondération sur chaque bande étroite [17]. Il est alors possible de regrouper toutes les bandes pour obtenir le modèle FSCK. Dans ce modèle, toutes les bandes étroites sont regroupées et les fonctions f et g obtenues de ce regroupement à une température donnée sont exprimées par :

$$f_{wb}(\kappa) = \frac{1}{\sigma T^4} \sum_{i=1}^{367} I_{b\nu,i} f_i(\kappa) \Delta\nu_i \quad (1.26)$$

où σ est la constante de Stefan-Boltzmann, $I_{b\nu,i}$ est la luminance du corps noir pour la bande étroite i à la fréquence ν . La même pondération est affectée à la fonction cumulée :

$$g_{wb}(\kappa) = \frac{1}{\sigma T^4} \sum_{i=1}^{367} I_{b\nu,i} g_i(\kappa) \Delta\nu_i \quad (1.27)$$

L'utilisation pratique de ce modèle se fait en plusieurs étapes nécessitant l'inversion de la fonction g . En effet, pour réaliser l'intégrale de l'équation 1.15, il faut connaître la valeur de la fonction H pour le coefficient d'absorption κ correspondant au point de quadrature g_i . On dispose au départ de la quadrature qui donne les valeurs de g_j et les poids w_j associés pour le spectre regroupé. Il faut alors chercher pour chaque valeur de g_j la valeur κ_j telle que l'équation 1.27 soit respectée. Pour une valeur de g_j , cette opération se fait par dichotomie. Pour chaque valeur de κ testée, on calcule donc la valeur de $g(\kappa)$ pour chaque bande, puis on calcule la valeur de g_{wb} en utilisant l'équation 1.27. La valeur maximale de départ prise pour κ_j est de 10 fois la valeur maximale du paramètre $\bar{\kappa}$ pour l'ensemble des bandes étroites. Un critère d'arrêt de la dichotomie à 50 itérations a été introduit afin d'éviter des phénomènes de boucle infinie.

L'inversion par dichotomie peut être consommatrice en temps de calcul et il sera donc souhaitable de rechercher une tabulation pour éviter cette opération. Toutefois, une fois cette opération effectuée, l'intégration spectrale se fait alors très rapidement, puisqu'il n'y a plus que N_{quad} calculs spectraux à effectuer.

Modèle WSGG

Dans le modèle de somme pondérée de gaz gris, l'émissivité totale ϵ_T d'une colonne de gaz de longueur l à la température T est donnée par :

$$\epsilon_T(T, l, p_a) = \sum_{k=1}^n a_k(T) [1 - \exp(-\kappa_k p_a l)] \quad (1.28)$$

où κ_k est le coefficient d'absorption du $k^{\text{ième}}$ gaz gris, p_a est la pression partielle des espèces absorbantes, et $a_k(T)$ est le facteur de pondération du $k^{\text{ième}}$ gaz. La pondération $a_k(T)$ peut être interprétée comme la fraction du rayonnement du corps noir dont le coefficient d'absorption est proche de κ_k . Dans le cas d'un milieu homogène et isotherme, la résolution de l'équation de transfert radiatif peut être remplacée par un système de n équations de transfert pour n gaz gris :

$$I = \sum_{k=1}^n I^k \quad (1.29)$$

$$\frac{dI^k}{ds} = \kappa_k p_a (a_k I_b - I^k) \quad (1.30)$$

Ce même système est couramment utilisé pour des milieux inhomogènes et anisothermes. Le facteur de pondération w_k est alors pris à la température locale.

Les paramètres du modèle de gaz gris qui sont le nombre de gaz gris n , les coefficients d'absorption κ_k et les pondérations $a_k(T)$ ont été calculés par de nombreux auteurs pour la vapeur d'eau seule, pour des mélanges $CO_2 - H_2O$, ou même en prenant en compte le rayonnement de la suie. dans le cas de mélanges $CO_2 - H_2O$, la tabulation doit être faite pour des valeurs données du rapport entre la pression partielle d'eau (p_w) et celle de CO_2 (p_c). Ces rapports de pressions $\frac{p_w}{p_c}$ sont couramment choisis égaux à 2, 1, ou 0.5. Le modèle choisi pour PRISSMA est un modèle à 3 gaz gris plus un gaz transparent, dont les paramètres ont été calculés pour des mélange $CO_2 - H_2O$ à pression atmosphérique et avec un rapport $\frac{p_w}{p_c} = 2$ correspondant par exemple à une situation de combustion méthane-air [12]. Ces paramètres ont été calculés à partir de données spectrales SNB couvrant une région allant de $150cm^{-1}$ à $7000cm^{-1}$. Elles ne couvrent donc pas la totalité du spectre utilisé dans le modèle SNB-CK, en particulier pour les grands nombres d'onde, mais prennent en compte les bandes les plus importantes pour H_2O et CO_2 .

Le modèle choisi comprenant 3 gaz gris, l'équation 1.28 se réécrit :

$$\epsilon_T(T, l, p_a) = \sum_{k=1}^n a_k(T) [1 - \exp(-\kappa_k p_w l)] \quad (1.31)$$

Et comme le quatrième gaz est un gaz transparent, la relation suivante lie les coefficients a_k :

$$a_4 = 1 - \sum_{k=1}^3 a_k \quad (1.32)$$

Lors du calcul de ces paramètres, les coefficients κ_k et a_k ont été ajustés pour une température $T = 1700K$. Pour les autres températures couvrant une plage de $300K$ à $2500K$, les valeurs de K_k sont fixées à celles obtenues pour $T = 1700K$ et les coefficients a_k sont ajustés pour chaque température par un polynôme à 5 coefficients :

$$a_k(T) = \sum_{j=0}^5 \alpha_{kj} T_j \quad (1.33)$$

La valeurs des paramètres $a_k(T)$ sont représentées sur la figure 1.2.

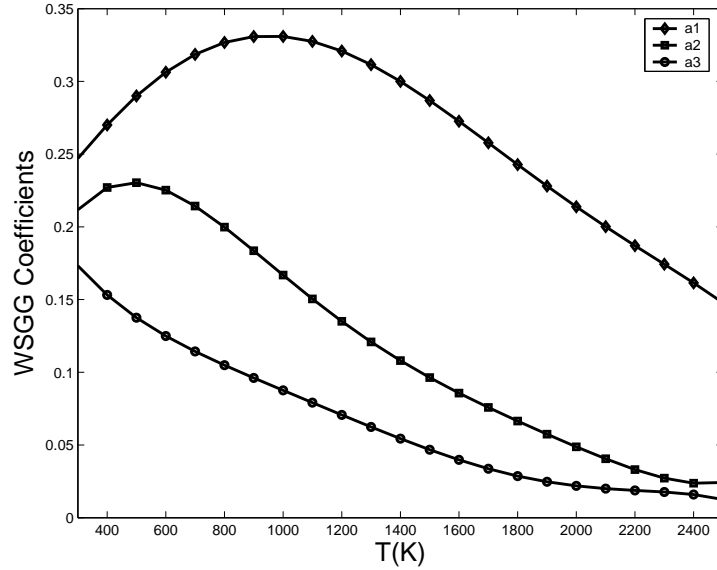


FIG. 1.2 – Coefficients pour le modèle WSGG

Chapitre 2

Installer le code

Dans ce chapitre, nous allons détailler toutes les étapes préparatoires afin de pouvoir utiliser le code PRISSMA. Dans un premier temps, nous verrons la configuration minimale requise pour pouvoir utiliser le code. Ensuite, nous expliquerons comment télécharger puis installer et compiler le code et toutes les routines qui l'accompagnent.

2.1 Configuration système requise

Le code a été compilé et testé sur des environnements de type Unix. Il doit être utilisable dans un environnement Windows, mais aucun test n'a été effectué.

2.1.1 Logiciels requis

Pour pouvoir télécharger et compiler le code ainsi que les routines qui l'accompagnent, il est nécessaire d'avoir les éléments suivants sur son système :

GCC (GNU Compiler Collection) : la version minimale nécessaire est la 4.2.3 ¹

Open MPI : une bibliothèque de fonctions qui permet de faire du calcul parallèle,

m4 : un processeur de macros, il permet de remplacer du texte comme nous le verrons dans la partie 2.4.2,

¹La routine `predatas` que l'on présente plus loin utilise une fonctionnalité de la norme Fortran 2003 : l'allocation dynamique dans un typé dérivé qui ne fut introduit qu'à partir de la version 4.2.x de GCC.

make : un logiciel qui permet d'exécuter les différentes opérations nécessaires à la compilation du code et des routines qui l'accompagnent.

Subversion (SVN) : un logiciel de suivi de version.

2.1.2 Logiciels recommandés

Outre les logiciels requis, certains autres peuvent s'avérer nécessaires pour tirer profit pleinement des possibilités offertes par PRISSMA afin d'effectuer le post-traitement des solutions obtenues :

Grace : un logiciel open source de tracé de courbes 2D,² ;

ParaView : une application open source d'analyse de données et de visualisation 3D³.

Nous verrons au chapitre 4 que les tests s'appuient sur Grace pour générer automatiquement le tracé des courbes.

2.1.3 Aide à l'installation

sur Linux

Sur Linux, la plupart de ces logiciels sont soit installés par défaut sur le système soit disponibles via les dépôts associés aux distributions. La plupart des distributions récentes fournissent en standard GCC 4.3 ; sur les plus anciennes, il faudra vérifier que la version fournie est compatible :

```
gcc --version
```

et si nécessaire, télécharger et compiler GCC 4.2 (ou 4.3) ou éventuellement mettre à jour votre distribution Linux vers une version plus récente.

sur OS X

Sur OS X, les choses ne sont malheureusement pas aussi simples. Dans un premier temps, il est nécessaire d'installer les outils de développement Apple disponible à l'adresse suivante : <http://developer.apple.com/mac/>⁴, ne serait-ce que pour accéder à la commande **make**.

Ensuite, le plus simple est d'installer GCC 4.3 et Open MPI en utilisant le projet MacPorts⁵. Une fois MacPorts installé, il est nécessaire d'ajouter

²<http://plasma-gate.weizmann.ac.il/Grace/>

³<http://www.paraview.org/>

⁴Pour accéder au téléchargement, il est nécessaire de se créer un compte gratuit sur le site d'Apple.

⁵<http://www.macports.org/>

les lignes suivantes au fichier `.bash_profile` qui se trouve à la racine de votre compte utilisateur :

```
export PATH=/opt/local/bin:/opt/local/sbin:/usr/local/bin:$PATH
export PATH=/opt/local/lib/openmpi/bin:/opt/local/include/openmpi:$PATH
export MANPATH=$MANPATH:/opt/local/share/man
export INFOPATH=$INFOPATH:/opt/local/share/info
```

Ensuite dans le terminal, exécutez la commande suivante pour installer GCC 4.3 et Open MPI :

```
sudo port install openmpi +gcc43
```

Le logiciel Grace est également disponible via MacPorts : `sudo port install grace`.

2.2 Téléchargement du code source

Le code est actuellement développé et maintenu par le CERFACS. Le projet s'appuie sur le logiciel Subversion pour gérer le suivi de version et permettre le téléchargement du code source. Le dépôt Subversion est à l'adresse suivante :

<http://www.cerfacs.fr/domasiumsvn/trunk>

Le code n'est pas en accès libre, il est nécessaire de prendre contact avec la personne en charge du code. Actuellement, il s'agit de Jorge Amaya que l'on peut joindre par courriel à l'adresse jorge.amaya@cerfacs.fr.

Utilisation de Subversion

La première fois que l'on veut récupérer une copie du code source, il faut effectuer l'opération SVN appelée `checkout`. On se place dans le répertoire dans lequel on veut effectuer le téléchargement et on lance la commande suivante :

```
svn co http://www.cerfacs.fr/domasiumsvn/trunk.
```

Ensuite pour mettre à jour notre copie, on effectue l'opération SVN appelée `update` depuis le répertoire où se situe notre copie :

```
svn update
```

2.3 Organisation du code

Le code source est organisée en cinq répertoires : HOSTS, SEQCODE, SOURCES, TOOLS, WORK.

HOSTS

Chaque sous-répertoire du répertoire HOSTS définit la configuration système d'une machine, notamment les compilateurs Fortran utilisés pour compiler le code séquentiel ou le code parallèle. On appellera par la suite ce sous-répertoire un « host ».

SEQCODE et SOURCES

Ces répertoires contiennent respectivement la version séquentielle et la version parallèle du code PRISSMA.

TOOLS

Ce répertoire contient les routines de pré et post-traitement. Leur utilisation sera détaillée au chapitre 3.

WORK

Ce répertoire contient les données spectrales des mélanges considérés et les données des modèles de quadrature.

2.4 Installation

2.4.1 Variables d'environnement

Trois variables d'environnement doivent être définis pour le bon fonctionnement du code :

DOMHOME donne le chemin vers le répertoire PRISSMA,

DOM_VERSION définit la version de PRISSMA : actuellement 3.X,

DOM_HOSTTYPE définit le type de machine sur lequel on est et correspond au nom du sous-répertoire dans le répertoire HOSTS de PRISSMA.

Selon le type de shell disponible sur votre machine, vous déclarerez ces variables de la manière suivante pour les shell de type « Bourne shell » :

```
export DOMHOME=chemin/vers/\code
export DOM_VERSION=numéro_de_version
export DOM_HOSTTYPE=nom_du_type de_host
```

et de la manière suivante pour les shell de type « C shell » :

```
setenv DOMHOME chemin/vers/\code
setenv DOM_VERSION numéro_de_version
setenv DOM_HOSTTYPE nom_du_type de_host
```

dans le fichier de configuration correspondant, par exemple `.bashrc` pour le shell Bash ou `.cshrc` pour le C shell.

Pour une configuration de type « C shell », il est possible d'utiliser le fichier `$DOM_HOME/HOSTS/dom.cshrc` pour définir le type de la machine utilisé à partir des types prédéfinis. Il suffit d'ajouter dans le fichier `.cshrc`, après la définition de la variable **DOM_HOSTTYPE** :

```
source $DOM_HOME/HOSTS/dom.cshrc
```

Si la machine ne correspond pas à un type prédéfini il faut ajouter un nouveau type dans le répertoire HOSTS.

2.4.2 Définition de son host

Un host contient deux fichiers : `dom.inc` et `makefile.h`.

Le fichier `dom.inc` contient essentiellement les valeurs utilisées pour les substitutions effectuées dans le code source par le processeur de macro m4. On peut ainsi modifier la précision des entiers et des réels dans le code. Il est toutefois recommandé de faire ce genre de modifications en toute connaissance de cause.

Le fichier `makefile.h` permet de définir les compilateurs Fortran utilisés pour le code parallèle (variable FF) et le code séquentiel (variable FFS) ainsi que les options de compilation.

Pour chaque nouvelle machine ou type de machine, il est conseillé de se créer un host. Pour cela, on duplique un autre host et on le nomme du nom de la machine et on adapte le contenu à sa configuration système. Comme nous l'avons vu précédemment, le nom de ce répertoire est utilisé pour définir la variable d'environnement **DOM_HOSTTYPE**.

Remarque : Lors de la définition d'un nouveau type il est nécessaire d'ajouter également ce type dans le fichier :

```
$DOM_HOME/TOOLS/EXTERNAL/ensightgold_writecbin.c
```

Les fichiers de visualisation sont écrits en binaire dont le format dépend de la machine utilisée⁶. Il faut donc ajouter une définition correspondant au « nouveautype » défini dans la section `/* Switch to BigEndian if necessary */` :

```
#elif defined (nouveautype)
#  define FWRITE fwrite_byteswap
#  define BYTESWAPIO TRUE
ou
#elif if defined (nouveautype)
#  define FWRITE fwrite
#  define BYTESWAPIO FALSE
```

selon si la machine est de type « Big Endian » ou « Little Endian ».

Attention : Le nom « nouveautype » utilisé ne doit pas contenir de tiret '-'.

2.5 Compilation

La compilation des divers programmes s'appuient comme nous allons le voir sur la commande `make`.

2.5.1 Compilation des outils

Chaque outil peut être compilé séparément, il suffit de se placer dans son répertoire et d'exécuter la commande :

```
make clean ; make
```

À la racine du répertoire `TOOLS`, on trouve deux scripts shell : `clean_all` et `compile_all` qui permettent respectivement de supprimer les compilations précédentes et de compiler tous les outils. On peut ainsi proprement compiler les outils en exécutant la commande :

```
clean_all ; compile_all
```

2.5.2 Compilation du code séquentiel

Ce code est peu optimisé, des tests ont montré que le code parallèle tournant avec un seul processus est environ deux fois plus rapide. Nous vous

⁶<http://fr.wikipedia.org/wiki/Endianness>

conseillons donc d'utiliser plutôt la version parallèle même sur une machine ne comprenant qu'un simple processeur. Toutefois si vous souhaitez utiliser cette version, la compilation s'effectue en se plaçant dans le répertoire SEQCODE et en lançant la commande :

```
make clean ; make
```

2.5.3 Compilation du code parallèle

De la même manière que pour le code séquentiel, il suffit de se placer dans le répertoire SOURCES et de lancer la commande :

```
make clean ; make
```

Si votre système est bien configuré, les compilations devraient se passer problème et nous allons pouvoir passer au chapitre suivant où nous allons voir comment effectuer un calcul avec PRISSMA.

Chapitre 3

Préparation et lancement d'un calcul

Dans ce chapitre, nous allons voir étape par étape comment préparer un calcul, le lancer et exploiter les résultats obtenus.

3.1 Présentation d'un exemple de cas test

Pour expliquer la procédure pour préparer et exécuter, nous allons nous appuyer sur un cas test : le cas d'un cylindre inhomogène et anisotherme. Le cylindre a une longueur $L = 1.2\text{m}$ et un rayon $R = 0.3\text{m}$. Ses parois sont noires et ont une température $T = 800\text{K}$ sauf la paroi située à $x = L$ qui est à 300K . La pression est égale à la pression atmosphérique. Les profils de température et de fractions molaires des espèces gazeuses H_2O et CO_2 sont donnés par :

$$\begin{aligned} T(z, r) &= 800 + 1200(1 - r/R)(z/L), \\ X_{\text{H}_2\text{O}}(z, r) &= 0.05 \left[1 - 2(z/L - 0.5)^2 \right] (2 - r/R), \\ X_{\text{CO}_2}(z, r) &= 0.04 \left[1 - 3(z/L - 0.5)^2 \right] (2.5 - r/R). \end{aligned} \quad (3.1)$$

Nous considérerons que le milieu ne contient pas de suie.

3.2 Préparation du calcul

3.2.1 Création du répertoire de travail

On commence par créer un répertoire pour le calcul dans lequel on se place, on exécute ensuite la routine `createdomcase` qui permet de prépa-

rer le répertoire de travail. Cette routine commence par créer deux liens symboliques vers les répertoires contenant les données spectrales de mélange (SPECTRAL) et les données des modèles de quadrature (QUADRA). Elle crée ensuite deux répertoires INFILES et OUTFILES où seront écrits respectivement :

- les fichiers d'entrée (avec une extension .IN) du calcul créés pendant le pré-traitement,
- et les fichiers de sortie (avec une extension .OUT), résultats de l'exécution du calcul.

Ces deux types de fichier sont des fichiers binaires, leur structure est détaillée sur les pages Web suivantes :

- <http://www.cerfacs.fr/prisma/formats.html> pour les fichiers d'entrée,
- <http://www.cerfacs.fr/prisma/outformats.html> pour les fichiers de sortie.

Et pour terminer, la routine `createdomcase` vérifie la présence du fichier de configuration du calcul `prisma.choices`. Si celui-ci n'est pas présent, un modèle de ce fichier est copié dans le répertoire, il faut alors adapter son contenu à la définition de notre calcul. Ce fichier est au format texte et se présente sous la forme suivante :

```
'./INFILES'      ! Path to *.in files
'./SPECTRAL'    ! Path to spectral data
'./OUTFILES'    ! Output path
SOLINIT         ! Data Status [ SOLINIT | RESTART ]
DMFS            ! Spatial discretization [ DMFS | STEP | EXPON ]
SNB-CK         ! Medium type [ GRAY | WSGG | FSCK | SNB-CK ]
5              ! Number of quadrature points
1.e-6          ! Convergence criterion for reflection
NO             ! Homogeneous system [ NO | YES ]
```

Le fichier permet de définir :

lignes 1 à 3 le chemin vers trois des répertoires créés justement par la routine `createdomcase`,

ligne 4 l'état des données, deux valeurs sont possibles SOLINIT si l'on initie un calcul et RESTART si l'on relance un calcul,

ligne 5 le modèle de discrétisation spatiale

Trois modèles sont disponibles :

- le schéma diamant en flux moyen ou DMFS (Diamond Mean Flux Scheme),
- le schéma step (STEP),

- le schéma exponentiel (EXPON).

Une étude a montré que le modèle DMFS donnait de meilleurs résultats, il est donc conseillé de privilégier ce modèle pour effectuer ses calculs.

ligne 6 le modèle spectral, Quatre modèles spectraux sont utilisables :

- le modèle GRAY (GRAY)
- le modèle WSGG (WSGG)
- le modèle FS-SNBck (FSCK)
- le modèle SNBck (SNB-CK)

ligne 7 le nombre de points utilisés lors du calcul de quadrature,

ligne 8 la valeur du critère de convergence si on utilise des parois grises,

ligne 9 l'homogénéité ou non du milieu, ce critère permet de simplifier l'exécution du calcul au moment de l'affectation des valeurs des grandeurs physiques aux cellules si le milieu est homogène.

Maintenant que le répertoire de travail est prêt, l'étape suivante consiste à lire le maillage.

3.2.2 Lecture du maillage

Actuellement, PRISSMA est capable de lire deux types de maillage : le format dit neutre du logiciel Gambit (le meilleur associé à Fluent) et le format de maillage AVBP ¹. Le code sait lire des maillages hybrides, mais les maillages non coïncidents tels que des maillages hexaèdres/tétraèdres ne sont pas supportés.

Pour lire les données du maillage, on utilise la routine `predatas`. Le script commence par vérifier la présence du fichier de configuration `predatas.choices` dans le répertoire de travail. Au cas où il ne le serait pas, le script en crée un modèle dont il faudra modifier le contenu et relancer `predatas`. Ce fichier est au format texte et se présente sous la forme suivante :

```
'./meshes/cyl_coarse.neu' ! Mesh file name
'./QUADRA'                ! Quadrature definition directory
2                          ! Mesh type (1=AVBP, 2=GAMBIT)
'./INFILES/'              ! Output path
SNDOM                     ! Quadrature type [SNDOM|FVM|TNDOM|PERSO]
4                          ! Quadrature order
2                          ! Number of directions if PERSO quadrature
1. 0. 0.                  ! 1st direction for PERSO quadrature
-1. 0. 0.                 ! 2nd direction ...
```

¹<http://pantar.cerfacs.fr/4-26334-The-AVBP-code.php>

Le fichier permet de définir :

ligne 1 le chemin vers le fichier de maillage ;

ligne 2 le chemin vers le répertoire contenant les données sur les modèles de quadrature ;

ligne 3 le type de maillage, on utilise 1 pour AVBP et 2 pour Gambit ;

ligne 4 le chemin vers le répertoire des fichiers d'entrée ;

ligne 5 le modèle de quadrature (à compléter) ;

ligne 6 l'ordre de la méthode SN pour fixer le nombre de directions, les valeurs possibles sont 2 pour S2, 4 pour S4, 6 pour S6, 8 pour S8, 9 pour LC11, 12 pour S12 ;

ligne 7 et suivantes si l'on choisit l'option PERSO pour le modèle de quadrature à la ligne 5, on définit à cette ligne le nombre de directions et leur vecteur (x,y,z) aux lignes suivantes.

À partir des paramètres lus dans ce fichier, le maillage est lu et un ensemble de fichiers d'entrée sont créés dans le répertoire INFILE. Ces fichiers contiennent la géométrie du maillage, les directions et leur chemin associé de parcours des cellules du maillage.

L'étape suivante consistera à lire la solution initiale issue d'un calcul de CFD.

3.2.3 Lecture de la solution initiale

Pour lire la solution initiale, il existe deux routines : `solinit` et `solinit_avbp`. La routine `solinit` permet d'utiliser une solution analytique telle que celle donnée dans la présentation du cas test (Eq. 3.1). Tandis que la routine `solinit_avbp` sait lire les résultats d'un calcul effectué avec le code AVBP.

avec `solinit`

Lorsqu'on utilise `solinit`, deux fichiers sont nécessaires : `solinit.choices` et `perso.inc`. Comme précédemment, si l'un des fichiers (ou les deux) manque, un fichier type est créé. Son contenu devra être adapté et la routine relancée.

Le fichier `solinit.choices` ne contient que le chemin vers le répertoire des fichiers d'entrée (dans notre cas INFILES) :

```
'./INFILES'          ! *.in directory
```

Le fichier `perso.inc` permet de définir les solutions analytiques pour les différents champs nécessaires pour définir le milieu. Pour le cas test présenté au début de ce chapitre, le fichier prend la forme suivante :

```

R_ref = 0.3
L_ref = 1.2
r = SQRT(z(i)*z(i)+y(i)*y(i))

T_n(i)      = 800.+1200.*(1.-r/R_ref)*(x(i)/L_ref)

P_n(i)      = 1.
XH2O_n(i)   = 0.05*(1.-2. *(z(i)/L_ref-0.5)**2)*(2.-r/R_ref)
XCO2_n(i)   = 0.04*(1.-3. *(z(i)/L_ref-0.5)**2)*(2.5-r/R_ref)
XCO_n(i)    = 0.
XO2_n(i)    = 0.2*(1.-XH2O_n(i)-XCO2_n(i)-XCO_n(i))
XN2_n(i)    = 0.8*(1.-XH2O_n(i)-XCO2_n(i)-XCO_n(i))
XSOOT_n(i)  = 0.

kscattering_n(i) = 0.
kabsorption_n(i) = 0.1

```

Les différents profils à renseigner sont donc :

- la pression $P_n(i)$,
- la température $T_n(i)$,
- les concentrations des espèces $XH2O_n(i)$, $XCO2_n(i)$, $XCO_n(i)$, $XO2_n(i)$, $XN2_n(i)$,
- la concentration en suie $XSOOT_n(i)$.

Le contenu de ce fichier est utilisé au sein d'une boucle sur les nœuds du maillage et permet donc d'exprimer la valeur de ces différentes grandeurs au nœud. Pour accéder aux coordonnées du nœud, on utilise $x(i)$, $y(i)$ et $z(i)$. Il est également possible d'utiliser les réels suivants pour déclarer des données du système étudié : r , d , x_ref , y_ref , z_ref , R_ref , L_ref , T_ref . Dans l'exemple, on utilise R_ref et L_ref pour définir respectivement le rayon et la longueur du cylindre, ainsi que r pour définir la position radiale du nœud.

D'un point de vue technique, il est fortement conseillé d'écrire le contenu de ce fichier entre la 7ème et la 73ème colonne (inclusive), pour des raisons de compatibilité avec certains compilateurs Fortran.

avec `solinit_avbp`

Pour utiliser `solinit_avbp`, un seul fichier est nécessaire : `solinit_avbp.choices`. Comme précédemment, si ce fichier n'est pas présent, la routine en crée une version dont il faut adapter le contenu, et il faut relancer `solinit_avbp`. Le contenu du fichier est le suivant :

```
'./asciifile.txt'      ! ascii solution file obtained with avbp2dom
'./INFILES'           ! path to the input files *.in
0.                    ! Soot concentration
```

Le fichier permet de définir :

ligne 1 le chemin vers un fichier texte qui contient les solutions d'un calcul AVBP sous une forme pré-définie. Ce fichier texte est obtenu à partir des solutions d'un calcul en utilisant la routine `avbp2dom`.

ligne 2 le chemin vers le répertoire des fichiers d'entrée (dans notre cas INFILES)

ligne 3 la concentration en suie

À partir des paramètres lus dans ce(s) fichier(s), la routine crée de nouveaux fichiers d'entrée.

L'étape suivante consistera à lire les conditions aux limites du domaine.

3.2.4 Lecture des conditions aux limites

La lecture des conditions des solutions s'effectue en utilisant la routine `initbc`. Deux fichiers sont nécessaires : `initbc.choices` et `clperso.inc`. Comme précédemment, si l'un des fichiers (ou les deux) manque, un fichier type est créé. Son contenu devra être adapté et la routine relancée.

Le fichier `initbc.choices` ne contient que le chemin vers le répertoire des fichiers d'entrée (dans notre cas INFILES) :

```
'./INFILES'          ! *.in directory
```

Le fichier `clperso.inc` permet de définir les conditions aux limites du domaine. Pour le cas test présenté au début de ce chapitre, le fichier prend la forme suivante :

```
IF (patch.eq.1) THEN
  T = 800.
  emissivity = 1.
```

```
ELSEIF (patch.eq.2) THEN
  T = 300.
  emissivity = 1.
ELSEIF (patch.eq.3) THEN
  T = 800.
  emissivity = 1.
ENDIF
```

Dans ce fichier, on définit la température et l'émissivité de la paroi pour chaque section des limites du domaine. Chaque section se voit attribuer un numéro qui correspond à sa position dans le fichier de définition du maillage et on peut se référer à la section 1 via le test :

```
IF (patch.eq.1) THEN
  T = 300.
  emissivity = 1.
ENDIF
```

On peut ajouter une nouvelle section, on ajoute une condition :

```
ELSEIF (patch.eq.patch_number) THEN
  T = profil de température en fonction de x, y et z
  emissivity = emissivity
```

Le code se sert de ces profils (ou valeurs fixes) pour attribuer la température et l'émissivité de chacune des faces appartenant aux parois. Pour définir un profil, il est possible de définir et utiliser les variables : `L_ref`, `R_ref`, `x_ref`, `y_ref`, `z_ref`, et également d'utiliser `x`, `y` et `z` qui représentent les coordonnées du centre de la face considérée.

D'un point de vue technique, il est fortement conseillé d'écrire le contenu de ce fichier entre la 7ème et la 73ème colonne (incluse), pour des raisons de compatibilité avec certains compilateurs Fortran.

Le pré-traitement est maintenant terminé, tous les fichiers d'entrée nécessaires au calcul ont maintenant été créés dans le répertoire `INFILE`. Le calcul est maintenant prêt à être lancé.

3.2.5 Conversions des fichiers d'entrée

Dans certaines occasions, il peut être nécessaire de transférer les fichiers d'entrée et de sortie. Par exemple, on peut effectuer le pré-traitement, le

calcul et le post-traitement sur des machines différentes. Or, les fichiers d'entrée et de sortie sont des fichiers binaires et dépendent donc de l'endianess des machines concernées. Pour passer d'une machine à l'autre il peut être alors nécessaire de convertir les fichiers. Pour cela, il existe deux routines : `inconvert` et `outconvert` qui permettent de convertir les fichiers d'entrée et les fichiers de sortie. Dans les 2 cas, on convertit les binaires sur la première machine en une archive au format tar gz contenant tous les fichiers d'entrée au format ascii. Ensuite, on transfère l'archive sur la seconde machine, et on effectue l'opération inverse, on convertit les fichiers ascii en binaire.

Pour les utiliser, il faut fournir un fichier de configuration, selon la routine : `inconvert.choices` ou `outconvert.choices`.

Le fichier `inconvert.choices`

Il se présente sous la forme suivante :

```
'./INFILES'          ! Prisma *.in infiles directory
'./infiles_txt'      ! Compressed file with txt data
1                    ! Conversion type [ 1:bin2ascii | 2:ascii2bin ]
```

ligne 1 le chemin vers le répertoire des fichiers d'entrée (dans notre cas INFILES);

ligne 2 le chemin vers l'archive sans l'extension tgz;

ligne 3 le sens de la conversion, on choisit 1 pour convertir de binaire vers ascii et 2 pour convertir de ascii vers binaire.

Le fichier `outconvert.choices`

Il se présente sous la forme suivante :

```
'./INFILES'          ! Prisma *.in infiles directory
'./OUTFILES'         ! Prisma *.out outfiles directory
'./outfiles_txt'     ! Compressed file with txt data
1                    ! Conversion type [ 1:bin2ascii | 2:ascii2bin ]
```

ligne 1 le chemin vers le répertoire des fichiers d'entrée (dans notre cas INFILES);

ligne 2 le chemin vers le répertoire des fichiers de sortie (dans notre cas OUTFILES);

ligne 3 le chemin vers l'archive sans l'extension tgz;

ligne 4 le sens de la conversion, on choisit 1 pour convertir de binaire vers ascii et 2 pour convertir de ascii vers binaire.

3.3 Lancement du calcul

Le lancement du calcul dépend de la version du code que vous souhaitez utiliser.

Calcul avec le code séquentiel Il suffit d'utiliser le script `run_seq` et le calcul s'effectue. Le script vérifie la présence du fichier `prissma.choices` et en cas d'absence, il crée une version dont il faudra adapter le contenu, puis il faudra à nouveau relancer `run_seq`.

Calcul avec le code parallèle Pour le code parallèle, l'équivalent du script `run_seq` a été écrit : `run_para` mais il n'est pas toutefois à l'heure actuelle pas intégré à la version actuelle. Ce script admet un argument facultatif qui permet de définir le nombre de processus que l'on souhaite utiliser pour effectuer le calcul. Si l'argument n'est pas défini, le script considère que le calcul doit être effectué avec un seul processus. Par exemple si l'on veut effectuer un calcul avec quatre processus, on tape la commande : `run_para 4`.

Toutefois, si ce script ne vous a pas été fourni, il faut alors taper la commande suivante :

```
mpiexec -n 4 $DOMHOME/HOSTS/$DOM_HOSTTYPE/mpiprissma_$DOM_...
VERSION.e_$DOM_HOSTTYPE
```

où 4 signifie que l'on veut utiliser quatre processus lors du calcul.

3.4 Traitement des résultats

Pour faciliter l'exploitation des résultats du code de calcul, deux routines ont été développées : `dom2ascii` et `visu`. Nous allons voir dans les sections suivantes

3.4.1 Création de fichiers ascii sur un rayon

La routine `dom2ascii` permet d'obtenir les valeurs des grandeurs de sortie selon un rayon. La routine crée un fichier pour chacune des grandeurs de sortie. Il est nécessaire de fournir le fichier `dom2ascii.choices` qui se présente sous la forme suivante :

```
'./INFILES/'      ! *.in files directory
'./OUTFILES/'     ! *.out files directory
0. 0. 0.          ! coordinates of the first point
```


Le fichier permet de définir :

ligne 1 le chemin vers le fichier du maillage ;

ligne 2 le type de maillage, on utilise 1 pour AVBP et 2 pour Gambit ;

ligne 3 le chemin vers le répertoire des fichiers d'entrée (dans notre cas INFILES) ;

ligne 4 le chemin vers le répertoire des fichiers de sortie (dans notre cas OUTFILES) ;

ligne 5 le chemin vers le répertoire des fichiers créés par la routine visu (dans notre cas VISU) ;

ligne 6 le type de fichier soit 1 pour la version binaire au format Enight soit 2 pour le fichier au format ascii.

Chapitre 4

Utilisation des tests

Un ensemble de tests a été développé afin de répondre à plusieurs objectifs. Une première partie est orientée utilisateurs et a pour but de tester son installation et ensuite d’appréhender l’utilisation du code de manière automatisée sur des cas d’étude type. La deuxième partie est orientée développeur et permet de tester la validité des modifications ou les performances.

4.1 Téléchargement

Les tests sont accessibles dans un dépôt Subversion par la commande suivante :

```
svn co http://www.cerfacs.fr/prissmasvn/branches/TEST_CASES.
```

Vous pouvez retrouver une aide rapide sur l’utilisation de Subversion à la section 2.2.

4.2 Présentation des configurations utilisées

Les différents tests s’appuient sur trois configurations que nous allons détailler dans cette partie.

4.2.1 le cube

Le cube a une longueur $L = 1\text{m}$. La pression est égale à la pression atmosphérique. La température du milieu se répartit en trois tranches, le bord gauche est à $T = 600\text{K}$, le centre à $T = 700\text{K}$ et le bord droit à $T = 800\text{K}$. Les parois sont noires et la température varie selon les faces. La face gauche est à $T = 600\text{K}$, la face droite à $T = 800\text{K}$; la température de

la moitié gauche des faces latérales est à $T = 666.6666\text{K}$ et la moitié droite à $T = 733.3333\text{K}$. La composition du milieu est de 10% pour H_2O , 5% pour CO_2 et 75% pour N_2 et des traces de suie ($1.10^{-7}\%$).

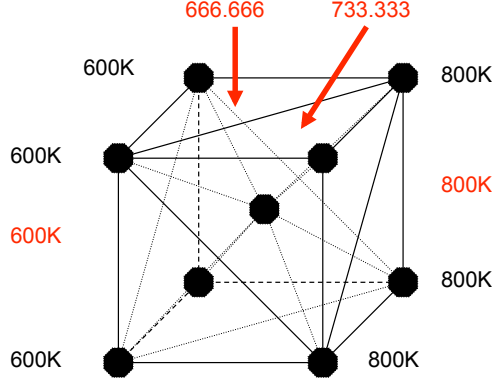


FIG. 4.1 – Configuration du cube.

4.2.2 le cylindre homogène et isotherme

Le cylindre a une longueur $L = 3\text{m}$ et un rayon $R = 0.5\text{m}$. La température du milieu est soit $T = 1200\text{K}$ soit $T = 1800\text{K}$ et la pression est égale à la pression atmosphérique. Les parois sont noires et ont une température $T = 300\text{K}$. La composition du milieu est de 20% pour H_2O , 10% pour CO_2 et 70% pour N_2 . Pour chaque température, deux cas sont envisagés : un cas sans suies et un cas avec des traces de suies ($1.10^{-7}\%$).

4.2.3 le cylindre inhomogène et anisotherme

Le cylindre a une longueur $L = 1.2\text{m}$ et un rayon $R = 0.3\text{m}$. Ses parois sont noires et ont une température $T = 800\text{K}$ sauf la paroi située à $x = L$ qui est à 300K . La pression est égale à la pression atmosphérique. Les profils de température et de fractions molaires des espèces gazeuses H_2O et CO_2 sont donnés par :

$$\begin{aligned} T(z, r) &= 800 + 1200(1 - r/R)(z/L), \\ X_{\text{H}_2\text{O}}(z, r) &= 0.05 \left[1 - 2(z/L - 0.5)^2 \right] (2 - r/R), \\ X_{\text{CO}_2}(z, r) &= 0.04 \left[1 - 3(z/L - 0.5)^2 \right] (2.5 - r/R). \end{aligned} \quad (4.1)$$

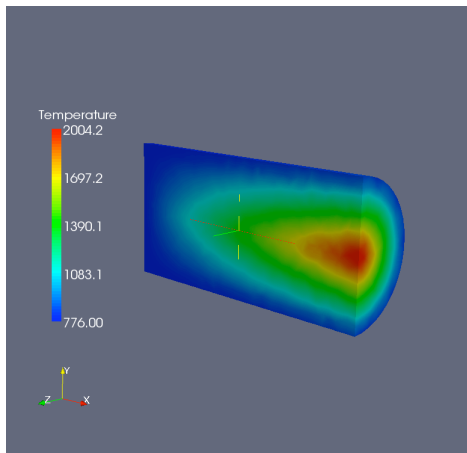


FIG. 4.2 – Champ de température le long de l'axe central.

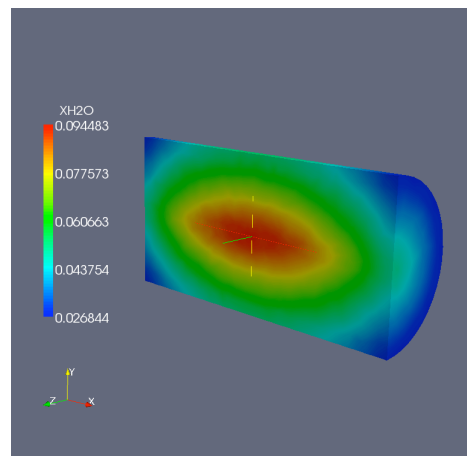


FIG. 4.3 – Champ de X_{H_2O} le long de l'axe central.

4.3 Les tests pour l'utilisateur

L'ensemble des tests pour l'utilisateur se situe dans le répertoire users. Le premier test présenté sert à vérifier que son installation du code PRISSMA s'est bien passé et que tout est en ordre. Les tests suivants ne sont pas vraiment des tests mais plutôt des études paramétriques où l'on va tester la validité des résultats calculés en les comparant à une référence numérique obtenue avec une autre méthode de calcul considérée comme plus précise, selon les cas : méthode de Monte-Carlo ou du Ray Tracing.

4.3.1 Tester son installation

Présentation

Un premier test a été développé pour tester son installation, il s'appuie sur une géométrie très simple : le cube. Le maillage réalisé pour le cube est très grossier (12 mailles uniquement) afin de pouvoir effectuer l'ensemble des calculs très rapidement : le temps de calcul est de l'ordre de la minute. Lors de l'exécution de ce test, un calcul est effectué pour chacun des modèles spectraux. Pour chaque modèle spectral, le résultat obtenu est comparé point par point avec une référence numérique (le résultat du même calcul validé sur une machine) le long d'un rayon. Si tout se passe bien, l'erreur mesurée doit être proche de zéro.

Si l'erreur obtenue est non négligeable, il faudra reprendre point par point l'installation et vérifier que tout a bien été fait selon les explications fournies

au chapitre 2.

Lancement du test

Pour effectuer ce test, il faut se placer dans le répertoire users et lancer la commande :

```
make TEST_INSTALL
```

4.3.2 Influence du modèle de discrétisation spatiale

Présentation

Dans cette étude paramétrique, on se place dans une configuration donnée, et on regarde l'influence du choix du modèle de discrétisation spatiale. Cette étude est disponible pour les deux configurations de cylindre. Pour chaque configuration, l'étude est menée pour deux modèles spectraux : le FS SNB-ck et le SNB-ck. Tous les calculs sont effectués avec la quadrature S4.

Pour le cylindre homogène et isotherme, nous utilisons la version isotherme à $T = 1200\text{K}$ sans trace de suie. Pour le cylindre inhomogène et anisotherme, nous utilisons la configuration sans suie.

Lancement des tests

Pour lancer les différentes variantes de cette étude paramétrique, il faut se placer dans le répertoire users des tests et lancer la commande :

```
make variante
```

où *variante* peut prendre les valeurs suivantes :

SPATIAL_DISCRETIZATION cette variante effectue tous les calculs

SPATIAL_DISCRETIZATION_HOMO cette variante effectue les calculs de variantes SPATIAL_DISCRETIZATION_HOMO_FSCK et SPATIAL_DISCRETIZATION_HOMO_SNB

SPATIAL_DISCRETIZATION_HOMO_FSCK cette variante effectue les calculs pour les trois modèles de discrétisation spatiale pour le cylindre homogène et isotherme à $T = 1200\text{K}$ sans trace de suie dans la configuration : quadrature S4 et modèle spectral FS SNB-ck

SPATIAL_DISCRETIZATION_HOMO_SNB cette variante effectue les calculs pour les trois modèles de discrétisation spatiale pour le cylindre homogène et isotherme à $T = 1200\text{K}$ sans trace de suie dans la configuration : quadrature S4 et modèle spectral SNB-ck

SPATIAL_DISCRETIZATION_INHOMO cette variante effectue les calculs de variantes **SPATIAL_DISCRETIZATION_INHOMO_FSCK** et **SPATIAL_DISCRETIZATION_INHOMO_SNB**

SPATIAL_DISCRETIZATION_INHOMO_FSCK cette variante effectue les calculs pour les trois modèles de discrétisation spatiale pour le cylindre inhomogène et anisotherme sans trace de suie dans la configuration : quadrature S4 et modèle spectral FS SNB-ck

SPATIAL_DISCRETIZATION_INHOMO_SNB cette variante effectue les calculs pour les trois modèles de discrétisation spatiale pour le cylindre inhomogène et anisotherme sans trace de suie dans la configuration : quadrature S4 et modèle spectral SNB-ck

Résultats

Les résultats concernant :

- le cylindre homogène et isotherme sont dans le répertoire `users/cyl_homo/results`
- le cylindre inhomogène et anisotherme sont dans le répertoire `users/cyl_inhomo/results`

4.3.3 Influence de la quadrature

Présentation

Dans cette étude paramétrique, on se place dans une configuration donnée, et on regarde l'influence du choix de la quadrature. Cette étude est disponible pour les deux configurations de cylindre. Pour chaque configuration, l'étude est menée pour deux modèles spectraux : le FS SNB-ck et le SNB-ck. Tous les calculs sont effectués avec la discrétisation spatiale DMFS.

Pour le cylindre homogène et isotherme, nous utilisons la version isotherme à $T = 1200\text{K}$ sans trace de suie. Pour le cylindre inhomogène et anisotherme, nous utilisons la configuration sans suie.

Lancement des tests

Pour lancer les différentes variantes de cette étude paramétrique, il faut se placer dans le répertoire `users` des tests et lancer la commande :

```
make variante
```

où *variante* peut prendre les valeurs suivantes :

QUADRATURE cette variante effectue tous les calculs

QUADRATURE_HOMO cette variante effectue les calculs de variantes **QUADRATURE_HOMO_FSCK** et **QUADRATURE_HOMO_SNB**

QUADRATURE_HOMO_FSCK cette variante effectue les calculs pour les six types de quadrature pour le cylindre homogène et isotherme à $T = 1200\text{K}$ sans trace de suie dans la configuration : quadrature S4 et modèle spectral FS SNB-ck

QUADRATURE_HOMO_SNB cette variante effectue les calculs pour les six types de quadrature pour le cylindre homogène et isotherme à $T = 1200\text{K}$ sans trace de suie dans la configuration : quadrature S4 et modèle spectral SNB-ck

QUADRATURE_INHOMO cette variante effectue les calculs de variantes **QUADRATURE_INHOMO_FSCK** et **QUADRATURE_INHOMO_SNB**

QUADRATURE_INHOMO_FSCK cette variante effectue les calculs pour les six types de quadrature pour le cylindre inhomogène et anisotherme sans trace de suie dans la configuration : quadrature S4 et modèle spectral FS SNB-ck

QUADRATURE_INHOMO_SNB cette variante effectue les calculs pour les six types de quadrature pour le cylindre inhomogène et anisotherme sans trace de suie dans la configuration : quadrature S4 et modèle spectral SNB-ck

4.3.4 Influence du modèle spectral

Présentation

Dans cette étude paramétrique, on se place dans une configuration donnée, et on regarde l'influence du choix du modèle spectral. Cette étude est disponible pour les deux configurations de cylindre. Pour chaque configuration, l'étude est menée avec ou sans trace de suie. Tous les calculs sont effectués avec le modèle de discrétisation spatiale DMFS et la quadrature S4.

Pour le cylindre homogène et isotherme, nous utilisons les deux versions d'isotherme : $T = 1200\text{K}$ et $T = 1800\text{K}$. Pour le cylindre inhomogène et anisotherme, nous utilisons la configuration sans suie.

Lancement des tests

Pour lancer les différentes variantes de cette étude paramétrique, il faut se placer dans le répertoire users des tests et lancer la commande :

```
make variante
```

où *variante* peut prendre les valeurs suivantes :

SPECTRAL cette variante effectue tous les calculs

SPECTRAL_HOMO cette variante effectue les calculs de variantes SPECTRAL_HOMO_1200K et SPECTRAL_HOMO_1800K

SPECTRAL_HOMO_1200K cette variante effectue les calculs pour les quatre modèles spectraux pour le cylindre homogène et isotherme à $T = 1200\text{K}$ avec et sans trace de suie dans la configuration : quadrature S4 et modèle de discrétisation spatiale DMFS.

SPECTRAL_HOMO_1800K cette variante effectue les calculs pour les quatre modèles spectraux pour le cylindre homogène et isotherme à $T = 1800\text{K}$ avec et sans trace de suie dans la configuration : quadrature S4 et modèle de discrétisation spatiale DMFS.

SPECTRAL_INHOMO cette variante effectue les calculs de variantes SPECTRAL_INHOMO_SOOT et SPECTRAL_INHOMO_NOSOOT

SPECTRAL_INHOMO_SOOT cette variante effectue les calculs pour les quatre modèles spectraux pour le cylindre inhomogène et anisotherme avec trace de suie dans la configuration : quadrature S4 et modèle de discrétisation spatiale DMFS.

SPECTRAL_INHOMO_NOSOOT cette variante effectue les calculs pour les quatre modèles spectraux pour le cylindre inhomogène et anisotherme sans trace de suie dans la configuration : quadrature S4 et modèle de discrétisation spatiale DMFS.

On obtient des figures comme les figures 4.4 et 4.5.

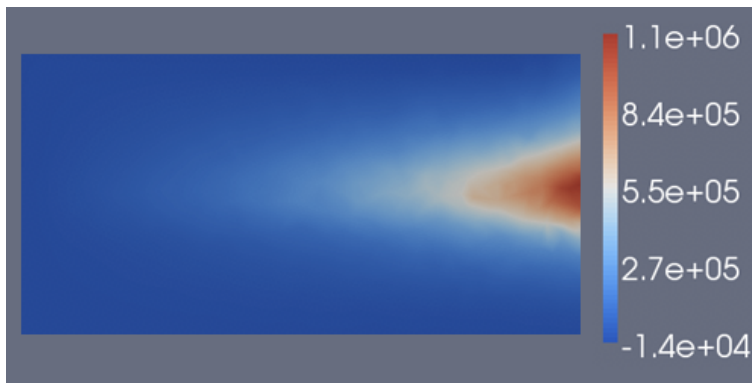


FIG. 4.4 – Profil du terme source radiatif S_r pour le cylindre inhomogène pour le modèle spectral SNB-ck (vue en tranche à $z=0$)

4.3.5 Remise à zéro

Au fur et à mesure que l'on effectue des tests, les différentes routines copient et créent des fichiers et accumulent des résultats dans le répertoire

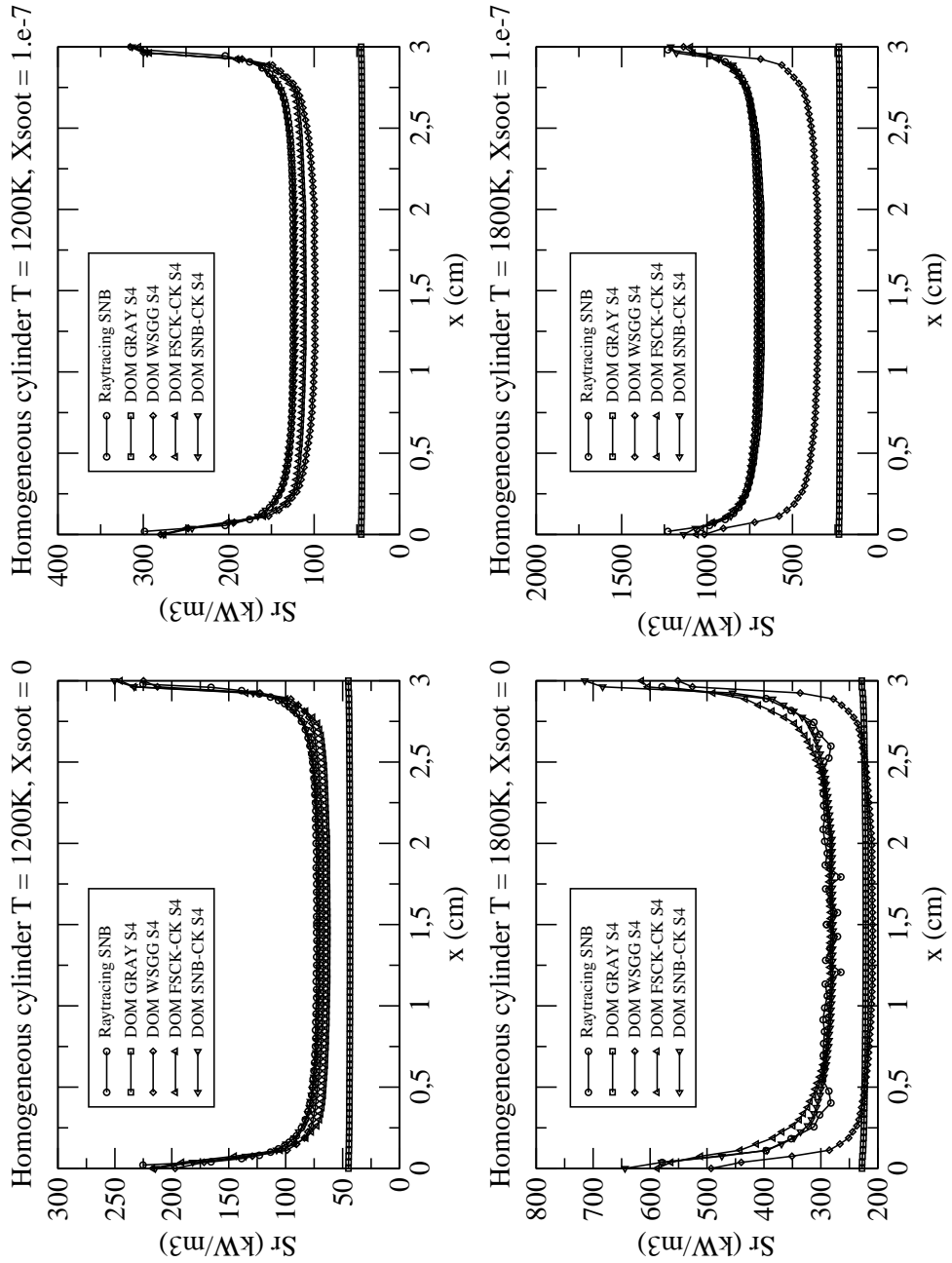


FIG. 4.5 – Valeur du terme source radiatif S_r dans les différents cas du cylindre homogène

results des trois configurations. Pour remettre à zéro ses répertoires, il suffit de se placer dans le répertoire users des tests et lancer la commande :

```
make CLEAN
```

cela remettra tous les répertoires dans leur état initial.

4.4 Les tests pour le développeur

L'ensemble des tests pour le développeur se situe dans le répertoire dev.

4.4.1 Validation numérique

Présentation

Ce test permet de valider les modifications apportées au code. Le principe est le même que celui défini dans la partie 4.3.1. Un calcul est effectué pour chacun des modèles spectraux. Pour chaque modèle spectral, le résultat obtenu est comparé point par point avec une référence numérique (le résultat du même calcul effectué dans une version du code qualifié) le long d'un rayon. Si tout se passe bien, l'erreur mesurée devrait être proche de zéro. En revanche, pour analyser plus finement les modifications, il s'appuie sur un cas plus concret que le cube : le cylindre inhomogène et anisotherme.

Lancement du test

Pour effectuer ce test, il faut se placer dans le répertoire compare et lancer la commande :

```
make COMPARE
```


Bibliographie

- [1] D. Joseph. *Modélisation des transferts radiatifs en combustion par méthode aux ordonnées discrètes sur des maillages non structurés tridimensionnels*. PhD thesis, Institut National Polytechnique de Toulouse, 2004. 6
- [2] D. Poitou. *Modélisation du rayonnement dans la simulation aux grandes échelles de la combustion turbulente*. PhD thesis, Institut National Polytechnique de Toulouse, Décembre 2009. 7
- [3] D. Joseph, M. El Hafi, R. Fournier, and B. Cuenot. Comparison of three spatial differencing schemes in discrete ordinates method using three-dimensional unstructured meshes. *Int. J. Thermal Sci.*, 44(9) :851–864, 2005. 7, 9, 10
- [4] K.A. Jensen, J.F. Ripoll, A.A. Wray, D. Joseph, and M. El Hafi. On various modeling approaches for radiative heat transfer in pool fires. *Combustion and Flame*, 148(4) :263–279, 2007. 7, 8, 13
- [5] S. Chandrasekhar. *Radiative transfer*. Clarendon Press, Oxford, 1950. 7
- [6] R. Koch and R. Becker. Evaluation of the quadrature schemes for the discrete ordinates method. In P. Lybaert, V. Feldheim, D. Lemonnier, and N. Selçuk, editors, *Proceedings of Eurotherm73 on Computational Thermal Radiation in Participating Media, Eurotherm Series 11*, pages 59–74, Paris, France, 2003. Elsevier. 8
- [7] J. Ströhle, U. Schnell, and K.R.G. Hein. A mean flux discrete ordinates interpolation scheme for general coordinates. In *3rd International Conference on Heat Transfer*, Antalya, Turkey, 2001. 9
- [8] J. Liu, H.M. Shang, Y.S. Chen, and T.S. Wang. Development of an unstructured radiation model applicable for two dimensional planar, axisymmetric and 3-dimensional geometries. *J. Quant. Spectrosc. Radiat. Transfer*, 66 :17–33, 2000. 10
- [9] A. Soufiani and J. Taine. High temperature gas radiative property parameters of statistical narrow band model for H_2O , CO_2 and CO and

- correlated-k model for H_2O and CO_2 . *Int. J. Heat Mass Transfer*, 40(4) :987–991, 1997. 10, 11
- [10] F. Liu, G.J. Smallwood, and Ö.L. Gülder. Application of the statistical narrow-band correlated-k method to low-resolution spectral intensity and radiative heat transfer calculations - effects of the quadrature scheme. *Int. J. Heat Mass Transfer*, 43 :3119–3135, 2000. 10, 12
- [11] F. Liu, G.J. Smallwood, and Ö.L. Gülder. Application of the statistical narrow-band correlated-k method to non-grey gas radiation in CO_2 - H_2O mixtures : approximate treatments of overlapping bands. *J. Quant. Spectrosc. Radiat. Transfer*, 68 :401–417, 2001. 10, 12
- [12] A. Soufiani and E. Djavdan. A comparison between weighted sum of gray gases and statistical narrow band radiation models for combustion applications. *Combustion and Flame*, 47(2) :240–250, 1994. 10, 15
- [13] D. Poitou, J. Amaya, C. Bhushan Singh, El Hafi M. Joseph, D., and B. Cuenot. Validity limits for the global model fs-snbck for combustion applications. In *Proceedings of Eurotherm83 - Computational Thermal Radiation in Participating Media III*, Lisbon, Portugal, 2009. 10
- [14] W. Malkmus. Random lorentz band model with exponential-tailed s-1 line-intensity distribution function. *Journal of Optical Society of America*, 57 :323–329, (1967). 11
- [15] Dufresne J-L., Fournier R., and Grandpeix J-Y. Inverse gaussian k-distributions. *J. Quant. Spectrosc. Radiat. Transfer*, 61, (1999). 11
- [16] F. Liu, G.J. Smallwood, and O.L. Gukder. Radiation heat transfer calculations using the snbck method. *33rd Thermophysics conference*, 1999. 13, 14
- [17] F. Liu, M. Yang, G.J. Smallwood, and H. Zhang. Evaluation of the snb based full-spectrum CK method for thermal radiation calculations in $\text{CO}_2 - \text{H}_2\text{O}$ mixtures. In *Proceedings of ICHMT 2004*, 2004. 14